<u>AMENDMENTS TO THE CLAIMS</u>

This listing of claims will replace all prior versions, and listings, of claims in the application:

<u>Listing of Claims:</u>

1.   (Currently amended) A system comprising:

a plurality of server nodes communicatively coupled on a network <u>of computer systems</u> to serve applications over the network to a plurality of clients<u>, the computer systems each having</u> <u>a processor and a memory and</u> the plurality of server nodes organized as instances, each instance including at least one server process <u>running in at least one of the computer system memories</u>;

a data object to store a hierarchical representation of configuration data associated with management of the server nodes, the data object accessible by all of the servers and the hierarchy having ~~a root,~~ a first subset of nodes ~~branching from the root~~ containing configuration data associated with all <u>of the</u> servers ~~in nodes branching from the root,~~ a second subset of nodes ~~branching from the root~~ containing configuration data specific to all servers of a first instance and a third subset of nodes ~~branching from a node of the second subset, the third subset of nodes~~ containing configuration data specific to a first server of the first instance;

a property sheet logically positioned at one of the nodes of the data object, the property sheet data structure including a plurality of property names, wherein each respective property name included in the property sheet data structure is associated with a default configuration value and, optionally, a custom configuration value to pair the default configuration value to the custom configuration value for <u>updating</u> a configuration parameter represented in the data object, wherein the property sheet data structure preserves both the ~~value of the~~ default ~~parameter~~ <u>configuration value</u> and the custom configuration value; and

a configuration manager on at least one of the server nodes to ~~determine whether the~~ <u>update</u> configuration data stored on the at least one server node ~~is out-of-date~~ based on the location of an updated configuration parameter within the hierarchy, wherein <u>all servers are to be</u>

updated in response to an updated configuration ~~parameters~~ parameter within the first subset of nodes ~~renders all servers out of date~~, all servers of the first instance are to be updated in response to an updated configuration ~~parameters~~ parameter within the second subset of nodes ~~renders all servers of the first instance out of date~~ and only the first server is to be updated in response to an updated configuration ~~parameters~~ parameter within the third subset of nodes ~~renders only the first server out of date~~.

2. (Currently amended) The system as in claim 1 wherein the data object is stored within a central database and wherein the hierarchy has a root, the first subset of nodes branches from the root, the second subset of nodes branches from the root and the third subset of nodes branches from a node of the second subset ~~accessible by each of the server nodes and a first node of the data object contains global configuration data associated with the plurality of server nodes and a second node of the data object contains configuration data specific to a one of the plurality of server nodes~~.

3. (Previously presented) The system as in claim 1 wherein, once the default configuration value has been modified, the default configuration value is stored independently with respect to the custom configuration values in the property sheet data structure.

4. (Previously presented) The system of claim 3, wherein the default configuration values associated with the property sheet data structure are modifiable using an interface other than a user interface.

5. (Previously presented) The system of claim 1, wherein the property sheet data structure is associated with a particular component or a set of components contained within a clustered system.

6. (Previously presented) The system of claim 1, wherein the property datasheet is modifiable with a user interface comprising:

a first dialog box to display contents of the property sheet data structure, the first dialog box including a plurality of entry rows, each respective entry row of the first dialog box including a first column to display names of corresponding properties, a second column to display current configuration values associated with corresponding properties and a third column to indicate if a configuration value displayed in the second column is a default configuration value or a custom configuration value; and

a second dialog box including a data entry field to enable a user to modify a selected default or custom value.

7. (Previously presented) The system of claim 6, wherein a custom configuration value associated with a property is modifiable by selecting the second dialog box of the corresponding property and entering a new value in the data entry field of the second dialog box.

8. (Previously presented) The system of claim 7, wherein the second dialog box of the corresponding property is selected by clicking a custom check box inside the third column of a corresponding entry row.

9. (Previously presented) The system of claim 8, wherein the second dialog box further includes a name field to display a name of a corresponding property and a default field to display a default configuration value associated with the corresponding property.

10. (Previously presented) The system of claim 9, wherein the second dialog box further includes a data type field to display the data type associated with corresponding property.

11.   (Currently amended) A method comprising:

storing within a <u>computer system memory a</u> data object[,] <u>providing</u> a hierarchical representation of configuration data associated with management of a plurality of server nodes organized as instances, each instance including at least one server process, the data object accessible by all of the servers and the hierarchy having ~~a root,~~ a first subset of nodes ~~branching from the root~~ containing configuration data associated with all <u>of the</u> servers ~~in nodes branching from the root,~~ a second subset of nodes ~~branching from the root~~ containing configuration data specific to all servers of a first instance and a third subset of nodes ~~branching from a node of the second subset, the third subset of nodes~~ containing configuration data specific to a first server of the first instance;

providing one or more property sheets at one or more of the nodes, each of the property sheets including a plurality of configuration parameters associated with the server nodes, each parameter associated with a name, a default configuration value and optionally a custom configuration value;

pairing the default configuration value with ~~an updated~~ <u>a</u> custom configuration value ~~for a~~ <u>to update one of the</u> configuration parameter<u>s</u> ~~represented in the data object~~ in response to ~~the~~ <u>a</u> user specifying a custom ~~parameter~~ value<u>,</u> wherein both the default configuration value and the custom configuration value are preserved by the property ~~data~~ sheet; and

determining whether to ~~invalidate~~ <u>update</u> the configuration data stored on ~~another~~ <u>one</u> of the server nodes based on the location of the updated configuration ~~parameters~~ <u>parameter</u> within the hierarchy, wherein ~~updated configuration parameters within the first subset of nodes invalidates~~ configuration data of all servers <u>is to be updated in response to an updated configuration parameter within the first subset of nodes,</u> ~~updated configuration parameters within the second subset of nodes invalidates~~ configuration data of all servers of the first instance <u>is to be updated in response to an updated configuration parameter within the second subset of nodes</u> and ~~updated configuration parameters within the third subset of nodes invalidates~~ configuration

data of only the first server <u>is to be updated in response to an updated configuration parameter</u> <u>within the third subset of nodes.</u>

12. (Currently amended) The method as in claim 11 further comprising:

storing the data object, configuration data, ~~binaries~~ and property sheets within a central database <u>and wherein the hierarchy has a root, the first subset of nodes branches from the root,</u> <u>the second subset of nodes branches from the root and the third subset of nodes branches from a</u> <u>node of the second subset</u>~~, the central database accessible by the server nodes and a first node of~~ ~~the data object containing global configuration data associated with the plurality of server nodes~~ ~~and a second node of the data object containing configuration data specific to a one of the~~ ~~plurality of server nodes~~.

13. (Currently amended) The method of claim 11, wherein specifying a custom ~~parameter~~ value in place of a default ~~parameter~~ value in a property sheet, comprises:

opening the property sheet in a property sheet graphical user interface, the graphical user interface comprising a first column for storing parameter names, a second column for storing a current parameter value and a third column for storing an indication as to whether the current parameter value is a custom value or a default value;

selecting the indication in the third column;

responsively generating a data entry window having a custom field for entering a custom value; and

receiving user entry of a custom value in the custom field.

14. (Original) The method as in claim 11 wherein the server nodes are Java server nodes supporting the Java 2 Enterprise Edition ("J2EE") standard and wherein the property sheet parameters comprise J2EE parameters.

15. (Currently amended) A method for updating configuration settings for a plurality of server nodes organized as instances, each instance including at least one server process, comprising:

storing, in a computer system memory, a data object providing a hierarchical representation of configuration data associated with management of the server nodes, the data object accessible by all of the servers and the hierarchy having ~~a root,~~ a first subset of nodes ~~branching from the root~~ containing configuration data associated with all of the servers ~~in nodes branching from the root,~~ a second subset of nodes ~~branching from the root~~ containing configuration data specific to all servers of a first instance and a third subset of nodes ~~branching from a node of the second subset, the third subset of nodes~~ containing configuration data specific to a first server of the first instance;

providing one or more property sheets at one or more of the nodes, each of the property sheets including a plurality of configuration parameters associated with the server nodes, each parameter associated with a name, a default configuration value and optionally a custom configuration value;

pairing the default configuration value with ~~an updated~~ a custom configuration value ~~for a~~ to update one of the configuration parameter~~s~~ ~~represented in the data object~~ in response to ~~the~~ a user specifying a custom ~~parameter~~ value, wherein both the default configuration value and the custom configuration value are preserved by the property ~~data~~ sheet;

communicating an indication of the configuration parameter update to one or more other server nodes;

determining ~~if the~~ whether to update configuration data stored on the other server nodes ~~is out of date~~ based on the location of the updated configuration ~~parameters~~ parameter within the hierarchy, wherein configuration data of all servers is to be updated in response to an updated configuration ~~parameters~~ parameter within the first subset of nodes ~~renders all servers out of date,~~ configuration data of all servers of the first instance is to be updated in response to an

updated configuration ~~parameters~~ parameter within the second subset of nodes ~~renders all servers of the first instance out-of-date~~ and <u>configuration data of only the first server is to be updated in response to an</u> updated configuration ~~parameters~~ parameter within the third subset of nodes ~~renders only the first server out-of-date~~; and

downloading the updated configuration ~~data~~ <u>parameter</u> from the central database to the other server nodes if <u>it is determined that</u> the configuration data stored on the other server nodes is ~~out-of-date~~ <u>to be updated</u>.

16.   (Previously presented) The method as in claim 15 further comprising:

acquiring a lock on the configuration parameters stored within the property sheet prior to updating the configuration parameters at the first server node.

17.   (Previously presented) The method as in claim 16 further comprising:

releasing the lock on the configuration parameters after the configuration data has been updated at the central database and/or at each of the other server nodes.

18.-22. (Cancelled)

23.   (New) A computer-readable medium having a set of instructions stored thereon which, when executed by a computer system, cause the computer system to perform a method comprising:

storing within a data object, a hierarchical representation of configuration data associated with management of a plurality of server nodes organized as instances, each instance including at least one server process, the data object accessible by all of the servers and the hierarchy having a first subset of nodes containing configuration data associated with all of the servers, a second

subset of nodes containing configuration data specific to all servers of a first instance and a third subset of nodes containing configuration data specific to a first server of the first instance;

　　providing one or more property sheets at one or more of the nodes, each of the property sheets including a plurality of configuration parameters associated with the server nodes, each parameter associated with a name, a default configuration value and optionally a custom configuration value;

　　pairing the default configuration value with a custom configuration value to update one of the configuration parameters in response to a user specifying a custom value, wherein both the default configuration value and the custom configuration value are preserved by the property sheet; and

　　determining whether to update the configuration data stored on one of the server nodes based on the location of the updated configuration parameter within the hierarchy, wherein configuration data of all servers is to be updated in response to an updated configuration parameter within the first subset of nodes, configuration data of all servers of the first instance is to be updated in response to an updated configuration parameter within the second subset of nodes and configuration data of only the first server is to be updated in response to an updated configuration parameter within the third subset of nodes.

　　24. (New) The computer-readable medium as in claim 23, wherein the method further comprises:

　　communicating an indication of the configuration parameter update to one or more other server nodes;

　　downloading the updated configuration parameter from a central database storing the hierarchical data object to the other server nodes if the configuration data stored on the other server nodes is out-of-date.

25. (New) The computer-readable medium as in claim 23, wherein specifying a custom value in place of a default value in a property sheet, comprises:

opening the property sheet in a property sheet graphical user interface, the graphical user interface comprising a first column for storing parameter names, a second column for storing a current parameter value and a third column for storing an indication as to whether the current parameter value is a custom value or a default value;

selecting the indication in the third column;

responsively generating a data entry window having a custom field for entering a custom value; and

receiving user entry of a custom value in the custom field.